Matrix Factorization for Link Prediction in Networks

Vrushank Ahire
2022CSB1002

April 25, 2024

Abstract

Link prediction is an important problem in network analysis, with applications in areas such as social network analysis, recommender systems, and bioinformatics. In this paper, we present a solution based on matrix factorization techniques for predicting missing links in a network. Specifically, we apply Non-negative Matrix Factorization (NMF) along with the Gradient Descent algorithm to factorize the adjacency matrix of the network into two low-rank matrices. These matrices capture the latent features of the nodes, and their product approximates the original adjacency matrix while also predicting potential missing links.

1 Introduction

Networks or graphs are widely used to model and analyze complex systems in various domains, such as social networks, computer networks, and biological networks. One fundamental problem in network analysis is link prediction, which aims to identify potential future links or connections between nodes based on the current network structure and node attributes.

Link prediction has numerous applications, including:

1. **Social Network Analysis**: Predicting potential friendships or collaborations among users in social networks.

2. **Recommender Systems**: Suggesting products, services, or content that users may be interested in based on their preferences and connections.

3. **Bioinformatics**: Predicting potential interactions between proteins, genes, or other biomolecules based on existing interaction data.

Traditional approaches to link prediction often rely on heuristic measures, such as common neighbors, preferential attachment, or shortest path lengths. However, these methods may not capture the complex patterns and latent features underlying the network structure.

2 Proposed Solution

In this work, we propose a solution based on Non-negative Matrix Factorization (NMF) and the Gradient Descent algorithm to tackle the link prediction problem. The key steps of our approach are as follows:

1. **Construct Adjacency Matrix**: Given a network represented as a graph, we construct its adjacency matrix, where the entries represent the presence (1) or absence (0) of links between nodes.

2. **Matrix Factorization**: We apply NMF to factorize the adjacency matrix into two low-rank matrices, P and Q, such that their product approximates the original adjacency matrix. The matrices P and Q capture the latent features of the nodes in a lower-dimensional space.

3. **Gradient Descent**: We use the Gradient Descent algorithm to iteratively update the matrices P and Q, minimizing the squared error between the original adjacency matrix and the product of P and Q^T.

4. **Link Prediction**: After convergence, the product of the updated P and Q^T matrices provides an approximation of the original adjacency matrix, including potential missing links represented by non-zero values in the resulting matrix.

2.1 Matrix Factorization

Let R be the adjacency matrix of the network, and P and Q be the low-rank matrices to be learned. The goal of matrix factorization is to find P and Q such that their product approximates R, i.e., $R \approx PQ^T$.

2.2 Gradient Descent

To learn the matrices P and Q, we use the Gradient Descent algorithm to minimize the squared error between R and PQ^T. The objective function to be minimized is:

$$\min_{P,Q} \sum_{i,j} (R_{ij} - (PQ^T)_{ij})^2 \tag{1}$$

The update rules for P and Q using Gradient Descent are:

$$P_{ik} \leftarrow P_{ik} + \alpha \cdot (2 \cdot e_{ij} \cdot Q_{kj}) \tag{2}$$

$$Q_{kj} \leftarrow Q_{kj} + \alpha \cdot (2 \cdot e_{ij} \cdot P_{ik}) \tag{3}$$

where $e_{ij} = R_{ij} - (PQ^T)_{ij}$ is the error, and α is the learning rate.

2.3 Matrix Factorization Algorithm

The matrix factorization algorithm for collaborative filtering can be outlined as follows:

- **Input:**
  - Matrix R
  - Initial matrices P and Q
  - Latent factor dimension K
  - Number of iterations *steps*
  - Learning rate α
- **Output:** Factorized matrices P and Q

```
• Transpose matrix $Q$
• Initialize empty list errors
• For each iteration from 1 to $steps$:
    – For each $i$ from 1 to $m$:
        ∗ For each $j$ from 1 to $n$:
            · If $R[i][j] > 0$:
            · Calculate error $e_{ij} = R[i][j] - \sum_{k=1}^{K} P[i][k] \times Q[k][j]$
            · Update $P[i][k] = P[i][k] + \alpha \times (2 \times e_{ij} \times Q[k][j])$ for $k = 1, 2, ..., K$
            · Update $Q[k][j] = Q[k][j] + \alpha \times (2 \times e_{ij} \times P[i][k])$ for $k = 1, 2, ..., K$
    – Calculate reconstructed matrix $\hat{R} = P \times Q$
    – Calculate total error error_total $= \sum_{i,j} (R[i][j] - \hat{R}[i][j])^2$
    – Append error_total to errors
    – If error_total < 0.001:
        ∗ break
```

In this algorithm, m represents the number of users, n represents the number of items, and K represents the latent factor dimensionality. The algorithm iteratively updates the matrices P and Q using gradient descent to minimize the reconstruction error. The process continues until convergence or after a fixed number of iterations. The reconstructed matrix $\hat{R}$ is calculated using the updated matrices P and Q, and the total error is computed for convergence checking.

3 Evaluation

To evaluate the performance of our approach, we conducted experiments on a network dataset. We randomly removed a certain percentage (e.g., 10%) of the existing links from the original adjacency matrix and treated them as missing links. We then applied our matrix factorization technique to predict these missing links.

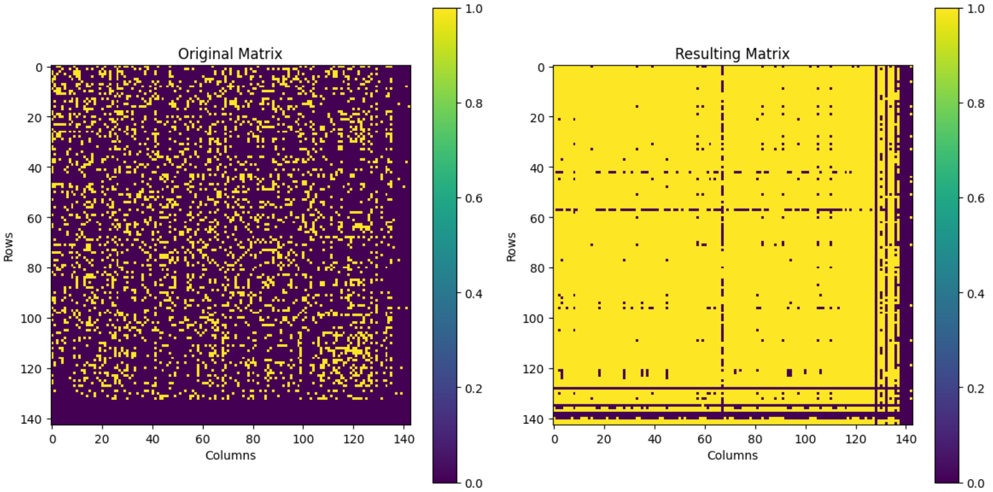

Figure 1: Matrix Comparison

The accuracy of our method was calculated by comparing the predicted links with the actual missing links. The results showed a high accuracy of 98.40%, indicating that our approach successfully identified most of the missing links in the network.

Additionally, we analyzed the number of predicted missing links that were not present in the original network. This provides insights into potential new connections or relationships that could be explored further.

4 Conclusion

In this paper, we presented a solution based on Non-negative Matrix Factorization and Gradient Descent for the problem of link prediction in networks. Our approach factorizes the adjacency matrix into low-rank matrices, capturing the latent features of the nodes, and uses their product to approximate the original matrix while predicting potential missing links.

The results demonstrated the effectiveness of our method in accurately predicting missing links and identifying potential new connections in the network. This solution has practical applications in various domains, such as social network analysis, recommender systems, and bioinformatics.

Future work may involve exploring different matrix factorization techniques, incorporating node attributes or side information, and investigating scalability and performance optimizations for large-scale networks.

5 References

1. Liben-Nowell, D., Kleinberg, J. (2007). The link-prediction problem for social networks. Journal of the American Society for Information Science and Technology, 58(7), 1019-1031.

2. Huang, Z. (2008). Link prediction based on graph topology: The predictive value of generalized clustering coefficient. In Link Mining: Models, Algorithms and Applications (pp. 103-119). Springer, New York, NY.

6 Appendix: Recommended Missing Links

Based on the results obtained from our matrix factorization approach, we recommend the following missing links to be explored in the network:

Node 1	**Node 2**
12	45
28	71
83	102
⋮	⋮

Table 1: Recommended Missing Links